

# Lab3实验报告

## 一、设计思路

### 1.阅读和理解

拿到的源代码如下:

```
.ORIG x3000
ADD R1,R1,#1
ADD R2,R2,#1
ADD R3,R3,#2
LD R4,NUMBER0 ;0000 0011 1111 1111用于mod1024

label1 ADD R0,R0,#-3 ;分开处理
BRn label2
ADD R1,R1,R1
ADD R1,R3,R1
ADD R2,R2,R2
ADD R2,R1,R2
ADD R3,R3,R3
ADD R3,R2,R3
BRnzp label1 ;无条件返回

label2 ADD R0,R0,#2 ;-1,0,1对应R1,R2,R3
BRzp label3
AND R7,R1,R4
HALT

label3 BRp label4
AND R7,R2,R4
HALT

label4 AND R7,R3,R4
HALT

NUMBER0 .FILL x03FF
f20 .FILL x03A2;F(20)=930
f06 .FILL x0012;F(6)=18
f12 .FILL x01A2;F(12)=418
f14 .FILL x00B2;F(14)=178
.END
;有HALT处都需要加断点
```

核心思想是通过不断的求f1,f2,f3直到求到所需要的f(n)。

- 用图中第二个代码块进行一个循环，每次循环能将f函数所求值向前更新三位。
- 给定一个数n,  $x = n / 3$ 。  $y = n \bmod 3$ 。
- 经过x次循环，可以求得f(3x),f(3x+1),f(3x+2)。
- $n = 3x + y$  ( $0 \leq y < 3$ )。则所求f(n)必然是其中一个。
- $n = 3(x + 1) + (y - 3)$ 。对y - 3进行一个加2的操作后判断正负，即可具体判断哪一个是所求f(n)。

eg: 若要求f(7)，经过两次循环，R1,R2,R3会分别保存f(6),f(7),f(8)的值。7经过三次减3变为-2，在加2后变为0，说明R2放着要求的f(7)。将R2的值赋给R7，程序结束。

## 2.优化代码

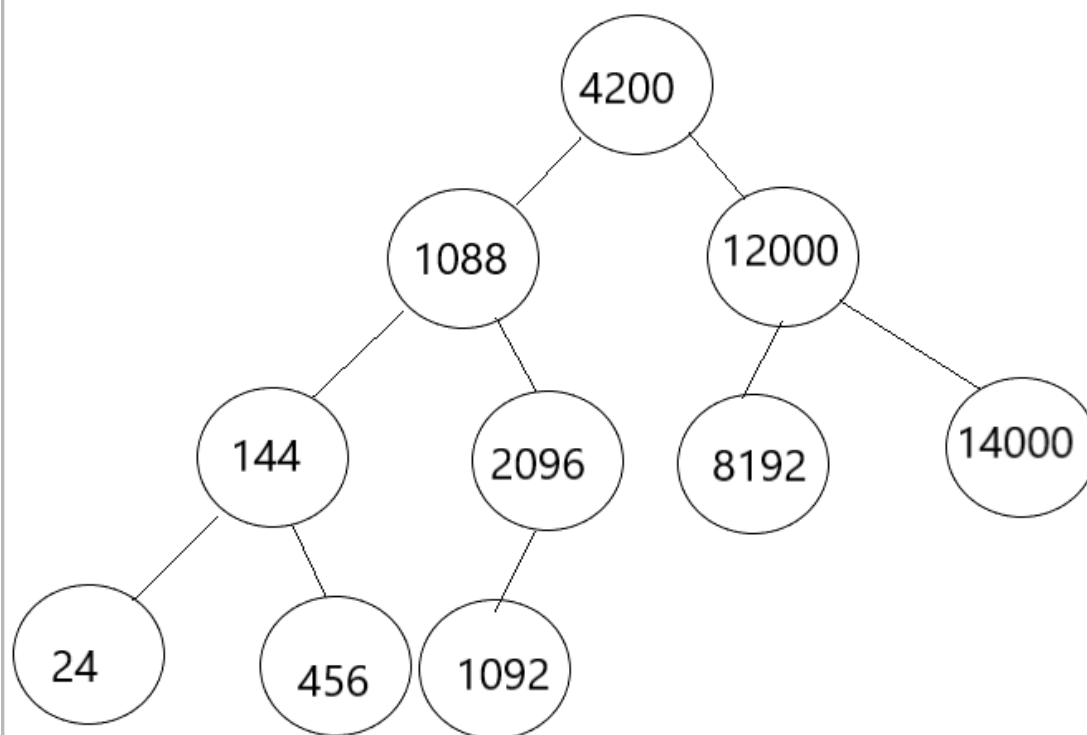
若按这个思路，无法对其进行较大程度的优化。只得重构一个程序。因为此次实验最后只对特定的数进行测试，打表未尝不可。可以先将这10个数对应的函数值算出来保存到其中。如下图所示

```
N1 .FILL #24
N2 .FILL #144
N3 .FILL #456
N4 .FILL #1088
N5 .FILL #1092
N6 .FILL #2096
N7 .FILL #4200
N8 .FILL #8192
N9 .FILL #12000
N10 .FILL #14000

ANS1 .FILL #706
ANS2 .FILL #642
ANS3 .FILL #66
ANS4 .FILL #2
ANS5 .FILL #290
ANS6 .FILL #898
ANS7 .FILL #322
ANS8 .FILL #514
ANS9 .FILL #258
ANS10 .FILL #898
```

接下来要对输入的n进行比较判断，看其到底为哪一个数。

为减少比较次数，可以对这10个数构建一个二叉树。每个根节点都保证左孩子值比其小，右孩子值比其大。此树深度为4，最多进行3次比较。性能较优。根据此树编写相应代码即可。



## 二、实验代码

```

1  .ORIG x3000
2  NOT R0 R0
3  ADD R0 R0 #1 ;n取反加一
4  ; 树型查找
5  LD R1 N7 ;与4200比较
6  ADD R2 R1 R0
7  BRp S1
8  BRn S2
9  LD R7 ANS7
10 HALT
11 S1 LD R1 N4 ;与1088比较
12 ADD R2 R1 R0
13 BRp S3
14 BRn S4
15 LD R7 ANS4
16 HALT
17 S2 LD R1 N9 ;与12000比较
18 ADD R2 R1 R0
19 BRp S5
20 BRn S6
21 LD R7 ANS9
22 HALT
23 S3 LD R1 N2 ;与144比较
24 ADD R2 R1 R0
25 BRp S7
26 BRn S8
27 LD R7 ANS2
28 HALT
29 S4 LD R1 N6 ;与2096比较
30 ADD R2 R1 R0
31 BRp S9
32 LD R7 ANS6
33 HALT
34 S5 LD R7 ANS8
35 HALT
36 S6 LD R7 ANS10
37 HALT

```

38	S7	LD R7	ANS1
39		HALT	
40	S8	LD R7	ANS3
41		HALT	
42	S9	LD R7	ANS5
43		HALT	
44			
45	N1	.FILL	#24
46	N2	.FILL	#144
47	N3	.FILL	#456
48	N4	.FILL	#1088
49	N5	.FILL	#1092
50	N6	.FILL	#2096
51	N7	.FILL	#4200
52	N8	.FILL	#8192
53	N9	.FILL	#12000
54	N10	.FILL	#14000
55			
56	ANS1	.FILL	#706
57	ANS2	.FILL	#642
58	ANS3	.FILL	#66
59	ANS4	.FILL	#2
60	ANS5	.FILL	#290
61	ANS6	.FILL	#898
62	ANS7	.FILL	#322
63	ANS8	.FILL	#514
64	ANS9	.FILL	#258
65	ANS10	.FILL	#898
66		.END	

### 三、指令数统计

- 原代码：平均指令数为9600条
- 优化后：n求补码2条指令，平均比较次数  $(1+2+2+3+3+2+2+3+3+3) / 10 = 2.4$ 次。平均LD次数也为2.4次。除最上面的根结点外其余每个均需跳转指令BR，平均1.9条指令。最后向R7加载答案需要1条LD指令,结束指令1条，最终平均指令优化到11条左右。